Cassandra Crossing/ L'insostenibile debolezza del middleware—log4j

(490)—Le vulnerabilità dell'ecosistema software globale non nascono dai codici delle applicazioni web, ma da oscure e comunissime…

Cassandra Crossing/ L'insostenibile debolezza del middleware—log4j
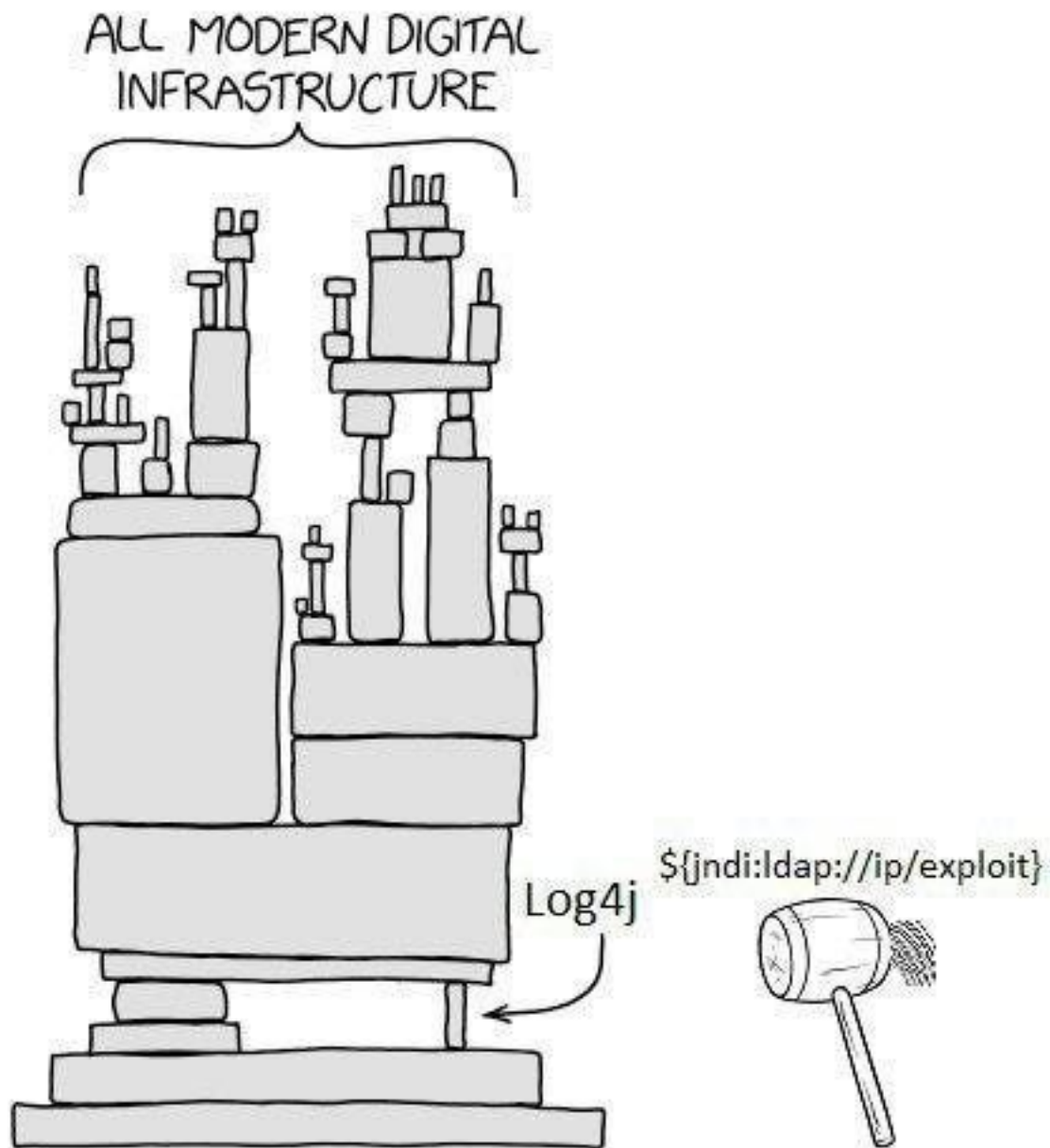
Figure 1:

(490)—Le vulnerabilità dell'ecosistema software globale non nascono dai codici delle applicazioni web, ma da oscure e comunissime librerie.

12 dicembre 2021—Come i 24 indefessi lettori di Cassandra sanno bene, la nostra profetessa preferita in passato si è occupata attivamente di sicurezza informatica.

La cosa più interessante che ricorda fu, nel lontanissimo 2011 o giù di lì ed in quel di Berlino, all'Universalhall-de, la partecipazione ad una sola (purtroppo) delle ultime edizioni di PH-Neutral, forse l' edizione 0x7db, conferenza hacker pubblica ed a pagamento, ma frequentabile solo con la "presentazione" di almeno due ex-partecipanti, disgraziatamente ormai estinta.

Ancora oggi devo ringraziare un paio di persone, anzi di carissimi amici ed amiche, che mi suggerirono di andare e mi "presentarono". Chissà se si ricordano ancora ....

Il nome PH-Neutral infatti esprime il concetto di "punto di equilibrio", luogo dove grey hat e white hat potevano incontrarsi tra pari.

Un evento memorabile, simile ad un CCC Camp condensato in poco più di 24 ore (birra, danze ed altro incluse).

A differenza del CCC il livello delle poche presentazioni (mattina e pomeriggio) era stratosferico (al CCC fanno anche molte "marchette"), e gli speaker erano di una bravura tale da risultare comprensibili a tutti, persino a Cassandra.

L'intervento che più mi colpì fu quello di un esperto di sicurezza nello sviluppo web (settore che allora cominciava ad esplodere) che spiegò, con esempi comprensibilissimi, il fatto che le prossime grandi vulnerabilità del web sarebbero derivate non dal codice scritto dai programmatori delle applicazioni stesse, ma da errori presenti negli infiniti pezzi di software che dovevano includere nelle applicazioni, librerie e "middleware".

Queste vulnerabilità sarebbero state sia veri e propri errori nel codice delle librerie stesse, sia errori di "interfacciamento" tra i diversi pezzi, dovuti ad API mal documentate o la cui documentazione, per la fretta, nessuno aveva il tempo di leggere a fondo.

Orbene, è notizia di questi giorni la scoperta di una vulnerabilità gravissima in un componente piccolo piccolo dei programmi Java; l'umile Log4j che come il suo fratellone Unix-like Syslog (a sua volta costola di Sendmail, ma finiamola qui) esiste per scrivere in file di testo cosiddetti "di log" quello che succede, soprattutto gli errori, durante l'esecuzione di un programma Java.

Si, i file di log, quegli "inutili" file che talvolta intasano i computer e gli smartphone, che servono per capire cosa non funziona quando l'aggeggio si pianta, e sono inoltre **pane e delizia per chi si occupa di Computer Forensics**.

Log4j è una libreria Java che svolge questo umile compito, e che è inclusa in quasi tutti i programmi Java, che a loro volta sono inclusi in quasi ogni servizio web al mondo.

Qui trovate la traduzione dal cinese di un buon articolo riassuntivo sulla questione

Come riassumere le possibili conseguenze della questione?

"Undici anni fa l'avevano già detto?"

"E' una questione così banale che qualunque programmatore da sempre sa che dobbiamo conviverci?"

Oppure, in maniera più originale ed accessibile, ed in tempi in cui la Cyberwar si è già materializzata (memento Stuxnet!) , possiamo affermare che "*tra la gente "normale", anche tra i "normali" addetti ai lavori, quasi nessuno si rende conto del pericolo, e di quanto possa essere facile* ***restare sconnessi, al buio ed all'asciutto come primo atto della prossima guerra****?*"

By Marco A. L. Calamari on December 13, 2021.

Canonical link

Exported from Medium on August 27, 2025.